bhist

Displays historical information about jobs

Synopsis

bhist [**-l** [**-aff**] [**-hostfile**]] [**-a**] [**-b**] [**-d**] [**-e**] [**-p**] [**-r**] [**-s**] [**-w**] [**-cname**] [**-app** application_profile_name] [**-C** start_time,end_ti
file_name | **-n** number_logfiles | **-n** min_logfile, max_logfile | **-n 0**] [**-S** start_time,end_time] [**-J** job_name] [**-Jd** "job_descripti
..."] [**-N** host_name | **-N** host_model | **-N** CPU_factor] [**-P** project_name] [**-q** queue_name] [**-u** user_name | **-u all** | **-G** user_gro

bhist -t [**-cname**] [**-f** logfile_name] [**-T** start_time,end_time]

bhist [**-h** | **-V**]

Description

By default:

* Displays information about your own pending, running, and suspended jobs. Groups information by job

* CPU time is not normalized

* Searches the event log file that is currently used by the LSF system: $LSB_SHAREDIR/*cluster_name*/logdir/lsb.events

* Displays events that occurred in the past week. Set the environment variable LSB_BHIST_HOURS to an alternative number of hours

Options

-a

Displays information about both finished and unfinished jobs.

This option overrides -d, -p, -s, and -r.

-aff

Displays historical job information about jobs with CPU and memory affinity resource requirement for each task in the job. If the job is pending, the requested affinity resources are displayed. For running jobs, the effective and combined affinity resource allocation is also displayed, along with a table headed AFFINITY that shows detailed memory and CPU binding information

for each task, one line for each allocated processor unit. For finished jobs (EXIT or DONE state), the affinity requirements for the job, and the effective and combined affinity resource requirement details are displayed.

Use only with the -l option.

-b

Brief format.

-cname

In LSF Advanced Edition, includes the cluster name for execution cluster hosts and host groups in output.

-d

Displays only information about finished jobs.

-e

Displays only information about exited jobs.

-hostfile

If a job was submitted with bsub -hostfile or modified with bmod -hostfile to point to a user-specified host file, bhist -l -hostfile shows the user-specified host file path. -hostfile also shows the contents of the host file.

-l

Long format.

If the job was submitted with bsub -K, the -l option displays Synchronous execution.

If you submitted a job using the OR (||) expression to specify alternative resources, this option displays the successful Execution rusage string with which the job ran.

If you submitted a job with multiple resource requirement strings using the bsub -R option for the order, same, rusage, and select sections, bhist -l displays a single, merged resource requirement string for those sections, as if they were submitted using a

single -R.

Long format includes information about:

* Job exit codes.

* Exit reasons for terminated jobs

* Job exceptions (for example, if job run time exceeds the runtime estimate, a job exception of runtime_est_exceeded displays)

* Resizable job information

* SSH X11 forwarding information (-XF)

* Specified and execution CWD. The full path is shown, including directory pattern values.

* Changes to pending jobs as a result of the following bmod options:

  * Absolute priority scheduling (-aps | -apsn)

  * Autoreszizable job attribute (-ar | -arn)

  * Current working directory (-cwd)

  * Post-execution command (-Ep | -Epn)

  * Job description (-Jd | -Jdn)

  * Checkpoint options (-k | -kn)

  * Migration threshold (-mig | -mign)

  * Job resize notification command (-rnc | -rncn)

* User limits (-ul | -uln)

* Runtime estimate (-We | -Wen)

-p
Displays only information about pending jobs.

-r
Only displays information about running jobs.

-s
Only displays information about suspended jobs.

-t
Displays job events chronologically, including energy aware scheduling events JOB_PROV_HOST and HOST_POWER_STATUS.

By default only displays records from the last week. For different time periods use -t with the -T option.

-w
Wide format. Displays the information in a wide format.

-app *application_profile_name*
Only displays information about jobs submitted to the specified application profile.

-C *start_time,end_time*
Only displays jobs that completed or exited during the specified time interval. Specify the times in the format yyyy/mm/dd/HH:MM. Do not specify spaces in the time interval string. This option overrides -r, -s, -p and -a.

For more information about the syntax, see "Time interval format" at the end of this bhist command reference.

-D *start_time,end_time*
Only displays jobs dispatched during the specified time interval. Specify the times in the format yyyy/mm/dd/HH:MM. Do not specify spaces in the

time interval string.

Must be used with -a option since it will only find results in running jobs.

For more information about the syntax, see "Time interval format" at the end of this bhist command reference.

-G *user_group*
Only displays jobs associated with a user group submitted with bsub -G for the specified user group. The -G option does not display jobs from subgroups within the specified user group.

The -G option cannot be used together with the -u option. You can only specify a user group name. The keyword all is not supported for -G.

-S *start_time,end_time*
Only displays information about jobs submitted during the specified time interval. Specify the times in the format yyyy/mm/dd/HH:MM. Do not specify spaces in the time interval string.

Must be used with -a option since it will only find results in running jobs.

For more information about the syntax, see "Time interval format" at the end of this bhist command reference.

-T *start_time,end_time*
Used together with -t.

Only displays information about job events within the specified time interval. Specify the times in the format yyyy/mm/dd/HH:MM. Do not specify spaces in the time interval string.

For more information about the syntax, see "Time interval format" at the end of this bhist command reference.

-f *logfile_name*
Searches the specified event log. Specify either an absolute or a relative path.

Useful for analysis directly on the file.

The specified file path can contain up to 4094 characters for UNIX, or up to 255 characters for Windows.

-J *job_name*
Only displays the jobs that have the specified job name.

The job name can be up to 4094 characters long. Job names are not unique.

The wildcard character (*) can be used anywhere within a job name, but cannot appear within array indices. For example job* returns jobA and jobarray[1], *AAA*[1] returns the first element in all job arrays with names containing AAA, however job1[*] will not return anything since the wildcard is within the array index.

-Jd "*job_description*"
Only displays the jobs that have the specified job description.

The job description can be up to 4094 characters long. Job descriptions are not unique.

The wildcard character (*) can be used anywhere within a job description.

-Lp *ls_project_name*
Only displays information about jobs belonging to the specified License Scheduler project.

-m "*host_name***...**"
Only displays jobs dispatched to the specified host.

-n *number_logfiles* **| -n** *min_logfile***,** *max_logfile* **| -n 0**
Searches the specified number of event logs, starting

with the current event log and working through the most recent logs in consecutive order. The maximum number of logs you can search is 100. Specify 0 to specify all the event log files in $(LSB_SHAREDIR)/*cluster_name*/logdir (up to a maximum of 100 files).

If you delete a file, you break the consecutive numbering, and older files are inaccessible to bhist. For example, if you specify 3, LSF searches lsb.events, lsb.events.1, and lsb.events.2. If you specify 4, LSF searches lsb.events, lsb.events.1, lsb.events.2, and lsb.events.3. However, if lsb.events.2 is missing, both searches include only lsb.events and lsb.events.1.

-N *host_name* **| -N** *host_model* **| -N** *cpu_factor*
Normalizes CPU time by the specified CPU factor, or by the CPU factor of the specified host or host model.

If you use bhist directly on an event log, you must specify a CPU factor.

Use lsinfo to get host model and CPU factor information.

-P *project_name*
Only displays information about jobs belonging to the specified project.

-q *queue_name*
Only displays information about jobs submitted to the specified queue.

-u *user_name* **| -u all**
Displays information about jobs submitted by the specified user, or by all users if the keyword all is specified. To specify a Windows user account, include the domain name in uppercase letters and use a single back slash (*DOMAIN_NAME*\fIuser_name) in a Windows command line or a double back slash (*DOMAIN_NAME\user_name*) in a UNIX command line.

job_ID **|** "*job_ID***[***index***]" ...**
Searches all event log files and only displays information about the specified jobs. If you specify a job array, displays all elements chronologically.

You specify job ID when you know exactly which jobs you want, so you should not specify any other options that control job selection (-a, -d, -e, -p, -r, -s, -D, -S, -T, -app, -G, -J, -Jd, -Lp, -M, -q, -u). If you specify an illogical combination of selection criteria, the system will not return any matching jobs.

In MultiCluster job forwarding mode, you can use the local job ID and cluster name to retrieve the job details from the remote cluster. The query syntax is:

bhist *submission_job_id@submission_cluster_name*

For job arrays, the query syntax is:

bhist "*submission_job_id*[*index*]"*@submission_cluster_name*

The advantage of using src_job_id@src_cluster_name instead of bhist -l job_id is that you can use src_job_id@src_cluster_name as an alias to query a local job in the execution cluster without knowing the local job ID in the execution cluster. The bhist output is identical no matter which job ID you use (local job ID or src_job_id@src_cluster_name).

You can use bhist 0 to find all historical jobs in your local cluster, but bhist 0@submission_cluster_name is not supported.

-h

Prints command usage to stderr and exits.

-V

Prints release version to stderr and exits.

Output: Default format

Memory Usage

Displays peak memory usage and average memory usage. For example:

MEMORY USAGE:

MAX MEM: 11 Mbytes; AVG MEM:6 Mbytes

You can adjust resource requirement accordingly next time for the same job submission if consumed memory is larger or smaller than current rusage.

Time Summary
Statistics of the amount of time that a job has spent in various states:.

PEND
The total waiting time excluding user suspended time before the job is dispatched.

PSUSP
The total user suspended time of a pending job.

RUN
The total run time of the job.

USUSP
The total user suspended time after the job is dispatched.

SSUSP
The total system suspended time after the job is dispatched.

UNKWN
The total unknown time of the job (job status becomes unknown if sbatchd on the execution host is temporarily unreachable).

TOTAL
The total time that the job has spent in all states; for a finished job, it is the turnaround time (that is, the time interval from job submission to job completion).

Output: Long format (-l)

The -l option displays a long format listing with the following additional fields:

Project
The project the job was submitted from.

Application Profile
The application profile the job was submitted to.

Command
The job command.

Detailed history includes job group modification, the date and time the job was forwarded and the name of the cluster to which the job was forwarded.

The displayed job command can contain up to 4094 characters for UNIX, or up to 255 characters for Windows.

Initial checkpoint period
The initial checkpoint period specified at the job level, by bsub -k, or in an application profile with CHKPNT_INITPERIOD.

Checkpoint period
The checkpoint period specified at the job level, by bsub -k, in the queue with CHKPNT, or in an application profile with CHKPNT_PERIOD.

Checkpoint directory
The checkpoint directory specified at the job level, by bsub -k, in the queue with CHKPNT, or in an application profile with CHKPNT_DIR.

Migration threshold
The migration threshold specified at the job level, by bsub -mig.

Requested Resources
Shows all the resource requirement strings you specified in the bsub command.

Execution CWD
The actual CWD used when job runs.

Host file
The path to a user-specified host file used when submitting or modifying a job.

Execution Rusage
This is shown if the combined RES_REQ has an rusage OR || construct. The chosen alternative will be denoted

here.

Effective RES_REQ

Displays a job's resource requirement as seen by the Scheduler after resolving any OR constructs.

Resizable job information

* For JOB_NEW events, bhist displays the auto resizable attribute and resize notification command in the submission line.

* For JOB_MODIFY2 events (bmod), bhist displays the auto resizable attribute and resize notification command in the submission line.

  * bmod -arn *jobID*:

    Parameters of Job are changed: Autoresizable attribute is removed;

  * bmod -ar *jobID*:

    Parameters of Job are changed: Job changes to autoresizable;

  * bmod -rnc *resize_notification_cmd* *jobID*:

    Parameters of Job are changed: Resize notification command changes to: <resize_notification_cmd>;

  * bmod -rncn *jobID*:

    Parameters of Job are changed: Resize notification command is removed;

* For JOB_RESIZE_NOTIFY_START event, bhist displays:

  Added <num_tasks> tasks on host <host_list>, <num_slots> additional slots allocated on <host_list>

* For JOB_RESIZE_NOTIFY_ACCEPT event, bhist displays the following:

* If the notification command is configured and sbatchd successfully initializes notification command. bhist displays

  Resize notification accepted. Notification command initialized (Command PID: 123456)

* If a notification command is not defined, bhist displays

  Resize notification accepted

* If sbatchd reports failure for whatever reason, bhist displays

  Resize notification failed

* For JOB_RESIZE_NOTIFY_DONE event, bhist displays the following:

  * Resize notification command completed if status is 0

  * Resize notification command failed if status is 1

* For JOB_RESIZE_RELEASE event, bhist displays

  Release allocation on <num_hosts> Hosts/Processors <host_list> by user or administrator <user_name>
  Resize notification accepted;

  For bmod -rncn, bhist displays

  Resize notification command disabled

* For JOB_RESIZE_CANCEL event, bhist displays

  Cancel pending allocation request

Synchronous execution

Job was submitted with the -K option. LSF submits the job and waits for the job to complete.

Terminated jobs: exit reasons

For jobs that have terminated, displays exit reasons.

Interactive jobs

For interactive jobs, bhist -l does NOT display information about a job's execution home, cwd, or running PID.

Dispatched <number> Task(s) on Host(s)

The number of tasks in the job and the hosts to which those tasks were sent for processing. Is displayed if**LSB_ENABLE_HPC_ALLOCATION** is set to Y or y in lsf.conf.

Allocated <number> Slot(s) on Host(s)

The number of slots that were allocated to the job based on the number of tasks, and the hosts on which the slots are allocated. Is displayed if**LSB_ENABLE_HPC_ALLOCATION** is set to Y or y in lsf.conf.

Requested Network and PE Network ID

Displays network resource requirements for IBM Parallel Edition (PE) jobs submitted with the bsub -network option, or to a queue (defined in lsb.queues) or an application profile (defined in lsb.applications) with the NETWORK_REQ parameter defined.

For example:

bhist -l 749
Job <749>, User <user1>;, Project <default>, Command <my_pe_job>

Mon Jun 4 04:36:12: Submitted from host <hostB>, to Queue <
priority>, CWD <$HOME>, 2 Tasks, Requested
Network <protocol=mpi:mode=US: type=sn_all:
instance=1:usage=dedicated>;
Mon Jun 4 04:36:15: Dispatched 2 Task(s) on Host(s) <hostb>,
Allocated <1> Slot(s) on Host(s) <hostb>;
Effective RES_REQ <select[type == local] rusage
[nt1=1.00] >, PE Network ID <1111111> <2222222>
used <1> window(s)per network per task;
Mon Jun 4 04:36:17: Starting (Pid 21006);

Output: Affinity resource requirements information (-l -aff)

Use -l -aff to display historical job information about CPU and memory affinity resource requirements for job tasks. A table w
taining the detailed affinity information for each task, one line for each allocated processor unit. CPU binding and memory bin
columns in the display.

HOST

The host the task is running on

TYPE

Requested processor unit type for CPU binding. One of
numa, socket, core, or thread.

LEVEL

Requested processor unit binding level for CPU binding.
One of numa, socket, core, or
thread. If no CPU binding level is requested, a
dash (-) is displayed.

EXCL

Requested processor unit binding level for exclusive CPU
binding. One of numa, socket, or core.
If no exclusive binding level is requested, a dash
(-) is displayed.

IDS

List of physical or logical IDs of the CPU allocation
for the task.

The list consists of a set of paths, represented as a
sequence integers separated by slash characters
(/), through the topology tree of the host. Each
path identifies a unique processing unit allocated to
the task. For example, a string of the form
3/0/5/12 represents an allocation to thread 12 in
core 5 of socket 0 in NUMA node 3. A string of the form
2/1/4represents an allocation to core 4 of socket
1 in NUMA node 2. The integers correspond to the node ID
numbers displayed in the topology tree from bhosts -aff.

POL

Requested memory binding policy.Eitherlocal or
pref. If no memory binding is requested, -
is displayed.

NUMA

ID of the NUMA node that the task memory is bound to. If

no memory binding is requested, a dash (-) is displayed.

SIZE

Amount of memory allocated for the task on the NUMA node.

For example the following job starts 6 tasks with the following affinity resource requirements:

```
bsub -n 6 -R"span[hosts=1] rusage[mem=100]affinity[core(1,same=socket,
exclusive=(socket,injob)):cpubind=socket:membind=localonly:distribute=pack]" myjob
Job <6> is submitted to default queue <normal>.

bhist -l -aff 6

Job <6>, User <user1>, Project <default>, Command <myjob>
Thu Feb 14 14:13:46: Submitted from host <hostA>, to Queue <normal>, CWD <$HO
                     ME>, 6 Task(s), Requested Resources <span[hos
                     ts=1] rusage[mem=100]affinity[core(1,same=socket,exclusive
                     =(socket,injob)):cpubind=socket:membind=localonly:distribu
                     te=pack]>;
Thu Feb 14 14:15:07: Dispatched 6 Task(s) on Host(s) <hostA> <hostA> <hostA>
                     <hostA> <hostA> <hostA>; Allocated <6> Slot(s) on Host(s)
                     <hostA> <hostA> <hostA> <hostA> <hostA> <hostA>;
                     Effective RES_REQ <select[type == local] order[r15s:pg]
                     rusage[mem=100.00] span[hosts=1] affinity [core(1,same=
                     socket,exclusive=(socket,injob))*1:cpubind=socket:membind=
                     localonly:distribute=pack] >;

AFFINITY:
                  CPU BINDING                    MEMORY BINDING
                  -----------------------        --------------------
HOST              TYPE   LEVEL  EXCL   IDS          POL   NUMA SIZE
hostA             core   socket socket /0/0/0       local 0    16.7MB
hostA             core   socket socket /0/1/0       local 0    16.7MB
hostA             core   socket socket /0/2/0       local 0    16.7MB
hostA             core   socket socket /0/3/0       local 0    16.7MB
hostA             core   socket socket /0/4/0       local 0    16.7MB
hostA             core   socket socket /0/5/0       local 0    16.7MB

Thu Feb 14 14:15:07: Starting (Pid 3630709);
Thu Feb 14 14:15:07: Running with execution home </home/jsmith>, Execution CWD
                     </home/jsmith>, Execution Pid <3630709>;
Thu Feb 14 14:16:47: Done successfully. The CPU time used is 0.0 seconds;
Thu Feb 14 14:16:47: Post job process done successfully;

MEMORY USAGE:
MAX MEM: 2 Mbytes;  AVG MEM: 2 Mbytes

Summary of time in seconds spent in various states by  Thu Feb 14 14:16:47
```

```
 PEND    PSUSP   RUN     USUSP   SSUSP   UNKWN   TOTAL
 81      0       100     0       0       0       181
```

Files

Reads lsb.events

See also

lsb.events: bgadd, bgdel, bjgroup, bsub, bjobs, lsinfo

Time interval format

You use the time interval to define a start and end time for collecting the data to be retrieved and displayed. While you can spe
also let one of the values default. You can specify either of the times as an absolute time, by specifying the date or time, or you
time.

Specify the time interval is follows:

start_time,*end_time*|*start_time*,|,*end_time*|*start_time*

Specify *start_time* or *end_time* in the following format:

[*year*/][*month*/][*day*][/*hour*:*minute*|/*hour*:]|.|.-*relative_int*

Where:

* *year* is a four-digit number representing the calendar year.

* *month* is a number from 1 to 12, where 1 is January and 12 is December.

* *day* is a number from 1 to 31, representing the day of the month.

* *hour* is an integer from 0 to 23, representing the hour of the day on a 24-hour clock.

* *minute* is an integer from 0 to 59, representing the minute of the hour.

* . (period) represents the current month/day/hour:minute.

* .-*relative_int* is a number, from 1 to 31, specifying a relative start or end time prior to now.

start_time,end_time
Specifies both the start and end times of the interval.

start_time,
Specifies a start time, and lets the end time default to now.

,end_time
Specifies to start with the first logged occurrence, and end at the time specified.

start_time
Starts at the beginning of the most specific time period specified, and ends at the maximum value of the time period specified. For example, 2/ specifies the month of February-start February 1 at 00:00 a.m. and end at the last possible minute in February: February 28th at midnight.

Absolute time examples

Assume the current time is May 9 17:06 2008:

1,8 = May 1 00:00 2008 to May 8 23:59 2008

,4 = the time of the first occurrence to May 4 23:59 2008

6 = May 6 00:00 2008 to May 6 23:59 2008

2/ = Feb 1 00:00 2008 to Feb 28 23:59 2008

/12: = May 9 12:00 2008 to May 9 12:59 2008

2/1 = Feb 1 00:00 2008 to Feb 1 23:59 2008

2/1, = Feb 1 00:00 to the current time

,. = the time of the first occurrence to the current time

,2/10: = the time of the first occurrence to May 2 10:59 2008

2001/12/31,2008/5/1 = from Dec 31, 2001 00:00:00 to May 1st 2008 23:59:59

Relative time examples

,.-2/ = the time of the first occurrence to Mar 7 17:06 2008

May 7 17:06 2008)